

2D1343, TENTAMEN I DATALOGI FÖR ELEKTRO
Onsdagen den 8 mars 2006 kl 14–19

Maxpoäng på tentamen är 50. Betygsgränser: 25 poäng ger trea, 35 ger fyra, 45 ger femma. Resultatet anslås senast 30 mars på CSCs anslagstavla.

Hjälpmedel: Ett A4-papper med dina egna handskrivna anteckningar och kursboken eller en algoritmbok och lexikon.

Kortfrågor

- (2p) 1. Vilken komplexitet har följande program-kod?
Kod i Python: Motsvarande kod i java:
- | | |
|---|---|
| <pre>def summera(inl): n = len(inl)+1 ul = [0]*n ul[0] = 0 i=1 while i < n: j=i-1 ul[i]=ul[j]+inl[j] i=i+1 return ul</pre> | <pre>int[] summera(int[] inl){ int n=inl.length+1; int[] ul=new int[n]; ul[0]=0; int i=1; while(i<n){ int j=i-1; ul[i]=ul[j]+inl[j]; i++; } return ul; }</pre> |
|---|---|
- (2p) 2. Rita en bild som visar hur en heap (träddform) ser ut när vi lägger in följande tal (i den ordning dem står) i en heape där tal med högst värde hamnar överst.
23, 43, 12, 45, 25, 42, 10
- (2p) 3. Anta att vi har ett fullt balanserat binärsökträd som innehåller heltal. Den enda information vi har om trädet är att höjden är 5 (d.v.s har 6 nivåer). Hur många tal finns i trädet? Hur många jämförelser behöver man som mest göra för att kunna avgöra om talet 4711 finns i trädet eller inte?

Problemfrågor

Stockholmsförsöket är ett försök för att minska trängseln, öka framkomligheten och förbättra miljön i Stockholm. Betalstationer som är utrustade med speciella kameror och sensorer har placerats ut runt omkring stan för att registrera in och utpassering med bil till och från stan.

4. *Sökning*
- (5p) Det finns information om 500000 passeringar i en lista. Listan är sorterad med avseende på registreringsnummer i alfabetisk ordning. Man vill kontrollera om registreringsnumret "BLX 212" finns i listan eller inte. Nämn och beskriv den sökalgoritm som passar bäst för att söka i den sorterade listan. Hur många jämförelser får man räkna med totalt innan man kan veta om "BLX 212" finns i listan eller inte?

5. Hashning

(6p)

Vi funderar på att lagra informationen om alla 500000 passeringar i en hashvektor. Hur stor hashvektor ska man välja? En hashfunktion har föreslagits som är summan av registreringsskyltens bokstävers ASCII-nummer gånger det tresiffriga talet. Är detta en lämplig hashfunktion? Beskriv två generella sätt att undvika krockar i hashvektorer.

6. Syntax

(8p)

Information som skickas från betalstationen ska följa en viss syntax för automatisk bearbetning. Följande är ett förslag på en syntaxform för meddelanden:

```
<info>      ::= BIL MED REGNR <regnr> HAR PASSERAT <station> KL <tid>
<regnr>     ::= <bkstv><bkstv><bkstv> <t2><t2><t2>
<station>   ::= SKANSBRON | SKANSTULLSBRON | LILJEHOLMSBRON
<tid>       ::= <timme>:<minut>
<timme>     ::= <t1><t2>
<minut>     ::= <t3><t2>
<t1>        ::= 0|1|2
<t2>        ::= 0|1|2|3|4|5|6|7|8|9
<t3>        ::= 0|1|2|3|4|5
<bkstv>     ::= A|B|C|D|E|F|G|H|J|K|L|M|N|O|P|R|S|T|U|W|X|Y|Z
```

Vilka av följande meningar följer *inte* syntaxen för <info>?

```
BIL MED REGNR DEM 123 HAR PASSERAT SKANSBRON KL 26:19
BIL MED REGNR MED 123 HAR PASSERAT LILJEHOLMSBRON KL 10:99
BIL MED REGNR RES 123 HAR PASSERAT HöGA-TORNET KL 01:59
BIL MED REGNR VIN 123 HAR PASSERAT SKANSTULLSBRON KL 12:10
```

Förklara vad det är som är fel i de felaktiga meningarna. Om man vill ha med sekundtalet också i klockslaget i denna form tt:mm.ss t.ex "11:23.24". Hur ska då syntaxen ändras?

7. *Sorteringsmetod*

(8p) När en bil passerar en betalstation läggs bilens registreringsnummer tillsammans med passeringsdatum, passeringstid och betalstationens namn i slutet av en lista. Listan kommer alltså att fyllas på kontinuerligt och därigenom automatiskt blir sorterad efter passeringstid. Listan kommer till slut att innehålla information om alla bilar som har passerat någon betalstation mellan klockan 06.30 till 18.30. Stockholms Stad anlitar dig för rådgivning om en sorteringsalgoritm för sortering av listan. Listan ska alltså sorteras efter bilarnas registreringsnummer men de ställer även krav på att information om passeringar för en viss bil är sorterad efter bilens passeringstider. Det är viktigt att komma med en tydlig, detaljerad och övertygande beskrivning av den sorteringsalgoritm som uppfyller kravet. (Algoritmen har bara tillgång till registreringsnumret, inte till passeringstiden.)

Listan före sortering

MDH 221	KTH 123	KTH 123	MDH 221	KTH 123
9:30		10:00		13:21		15:00		17:49	

Listan efter sortering med en sorteringsmetod som uppfyller kravet

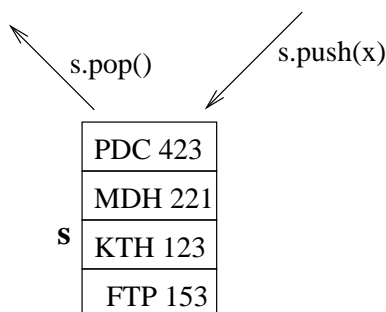
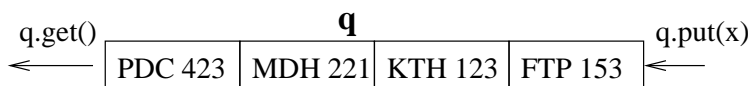
.....	KTH 123	KTH 123	KTH 123	MDH 221	MDH 221
	10:00	13:21	17:49		9:30	15:00	

Listan efter sortering med en sorteringsmetod som *inte* uppfyller kravet

.....	KTH 123	KTH 123	KTH 123	MDH 221	MDH 221
	17:49	10:00	13:21		15:00	9:30	

8. *Rekursiv flyttning*

(6p) Alla registreringsnummer finns i en abstrakt kö som endast har funktionerna `q.put(x)`, `q.get()` och `q.isEmpty()`. Vi har tillgång till en stack med funktionerna `s.push(x)` och `s.pop()`. Vi vill flytta alla registreringsnummer från kön till stacken, men det är viktigt att ordningen inte förstörs efter flyttningen. Alltså första registreringsnumret i kön ska hamna överst på stacken, nästa registreringsnummer i kön hamnar precis under det översta registreringsnumret på stacken osv. Detta ska göras med hjälp av funktionsanropet `flytta(q,s)`. Beskriv en rekursiv algoritm för funktionen `flytta` som flyttar alla registreringsnummer från kön `q` till stacken `s` med ovanstående krav uppfyllt.



9. *Tolkning av bilder*

Stockholmsförsöket använder ett OCR-program (Optical Character Recognition) för tolkning av kamerabilder till tecken. OCR-programmet klarar inte av att tolka alla tecken i ett registreringsnummer på bilder med dålig kvalitet. Vissa bilder måste därför tolkas manuellt. För varje bild som skickas till OCR-programmet får man ett tal och en sträng. Talet talar om hur många tecken som gick att tolka och strängen innehåller tolkade tecken samt tecknet # för de tecken som inte gick att tolka. Om tolkningen lyckas läggs registreringsnumret i en hashvektor.

(6p)

Exempel:

talet=6, strängen="AGA 912" (lyckad tolkning)

talet=5, strängen="AGA 9#2" (misslyckad tolkning)

Eftersom manuell tolkning tar lång tid räknar man med att en del misslyckade registreringsnummer inte kommer att hinna tolkas i tid. Därför vill man börja med att tolka de registreringsnummer som har färre misslyckade tecken (d.v.s. få antal # i strängen). Dessutom har erfarenhet visat att det går snabbare att tolka ett misslyckat tecken om tecknet är en siffra, därför vill man att registreringsnummer med fel i sifferdelen ska prioriteras före dem som har fel i bokstavsdelen. I kursen datalogi har vi gått igenom en datastruktur som skulle kunna passa bra för detta ändamål. Nämn och förklara hur datastrukturen fungerar. Om det redan finns något registreringsnummer i hashvektorn som skulle kunna passa vill man gärna få veta det. Förklara hur sökningen går till om talet är 5, d.v.s. bara ett # i strängen.

10. *Abstrakt passering*

Den som programmerat för stockholmsförsöket skryter med att hon använt en abstrakt datatyp för passeringarna. Vad finns det för fördelar med det? Föreslå metoder (minst tre) som kommer att behövas.

(5p)

Glöm inte klicka på kursvärderingsfrågor på kursens webbsida.